Course: ECE372A
Author:
NetID:
Date:
Project Name: Lab2

Description
In this lab, you will create a millisecond delay using Timer 0 to
control the rate at which several LEDs blink. These LEDs will
represent a binary number and will count up continuously to 15. A
switch will toggle the rate at which the binary number counts up
between an original rate and twice the original rate. The switch
will do so using an interrupt service routine.

Instructions
Examine comments in switch.cpp, main.cpp, led.cpp, and timer.cpp and
complete the described functions. You will need to create a circuit
using your breadboard, jumper wires, LEDs, switch and resistors to
control the LEDs and switch using the microcontroller.

Requirements
Overall
1. The program must compile. This can be tested using the checkmark
icon.
2. Program must be written following good coding protocols and be well
commented.
3. Arduino libraries are not allowed at all for this lab with the
exception of debug functionality using Serial.println.

switch.cpp
1. A function implementation that returns void and has no parameters
called initSwitchPB3 must be present and is used in the main function
to initialize the switch on the pin named *PB3*.
2. PB3 must be initialized with an input pull-up resistor
3. Pin-Change Interrupts must be enabled for pin PB3

led.cpp
1. A function implementation that returns void and has no parameters
called initLED must be present and is used in the main function to
initialize all LED pins as outputs.
2. Pins named PA0, PA1, PA2, and PA3 must be used to control the LEDs
3. A function implementation called turnOnLEDWithChar that returns
void and has a parameter called num of type unsigned unsigned char
must be present.
4. The turnOnLEDWithChar function must be **one line of code.** See
LED control section for more details.

switch.cpp
1. A function implementation that returns void and has no parameters
called initSwitch must be present and is used in the main function to
initialize the pin called *PB3* on the development board as an input.

timer.cpp
1. A function implementation that returns void and has no parameters called initTimer0 must be present and must initialize timer 0.
2. A function implementation that returns void and has no parameters called delayMs must be present and must implement a precise millisecond delay and can work at least up to 100 milliseconds. For loops are allowed.

main.cpp
1. A call to initLED() and initSwitchPB3() must be present in the main function.
2. An infinite while loop must be present.
3. A state machine for the entire project must be implemented using a typedef enum for states.
4. An ISR to handle the switch being pressed must be present and be used to change states appropriately.
5. The switch press must be debounced using an appropriately designed state machine.
6. When the switch is **pressed and released**, the LEDs change the pace at which they blink (either half or twice depending on the state).
7. LEDs must blink either every 100 ms or 200 ms depending on the state.

LED Control
The purpose of the function turnOnLEDWithChar is to practice a more advanced use of bit operations. We have covered how to manipulate a single bit at a time, however, there are times that are useful to manipulate several at a time.

Since the LEDs must be controled with PA0, PA1, PA2, and PA3, we can manipulate a single register and simulatenously manipulate several bits. The method to do this is to first "mask" these bits in a register by setting them to zero and then ORing them with the bits we want. An example of doing this in two steps is:

```
PORTA = PORTA & 0xF0; // sets the first four bits to zero.
PORTA = PORTA | (num & 0x0F); // sets the top four bits of num to zero. Then these bits are ORed with PORTA.
```

This can (and must for this lab) be done in one line of code.